Stability and Adaptability of Autonomous Decentralized Flow Control in High-Speed Networks

Chisa Takano
Traffic Engineering Division
NTT Advanced Technology Corporation
2-4-15, Naka-cho, Musashino-shi 180-0006, Japan
chisa@m.ieice.org

Masaki Aida
NTT Information Sharing Platform Laboratories
NTT Corporation
Musashino-shi 180-8585, Japan
aida@m.ieice.org

Abstract

This paper focuses on flow control in high-speed networks. Each node in a network handles its local traffic flow only on the basis of the information it is aware of, but it is preferable that the decision-making of each node leads to high performance of the whole network. To this end, we investigate the relationship between the flow control mechanism of each node and network performance. We consider the situation in which the capacity of a link in the network is changed but individual nodes are not aware of this. Then we investigate the stability and adaptability of the network performance when the capacity of a link is changed, and discuss an appropriate flow control model on the basis of simulation results.

1 Introduction

In a high-speed network, propagation delay becomes the dominant factor in the transmission delay because the speed of light is an absolute constraint. Therefore, at any given time, a large amount of data is being propagated on links in the network (Fig. 1). The amount of this data is characterized by the *delay-bandwidth product*, *i.e.*, the propagation distance multiplied by the transmission rate. Therefore, in high-speed and/or long-distance transmission, there is more data in transit on the links than there is in the nodes.

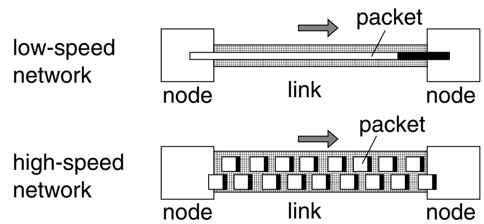

Figure 1. Effect of large delay-bandwidth product

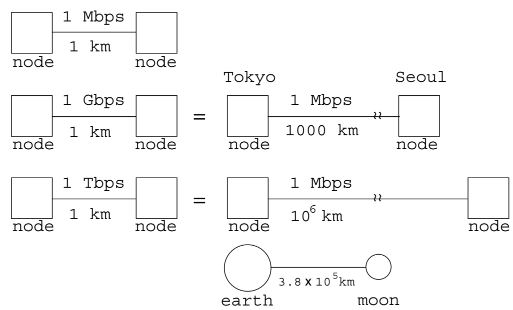

Figure 2. Example of delay-bandwidth product

Figure 2 shows an example of how much data there can be on a link. Let us consider the situation involving data transmission between two nodes, a distance of 1 km apart with a link speed of 1 Mbps. If the transmission speed is increased to 1 Gbps, the amount of data on the link is equivalent to that on 10^3 km of a 1-Mbps link. And, if the transmission speed is increased to 1 Tbps, the data volume is equivalent to 10^6 km of a 1-Mbps link. This distance is about 2.5 times the distance between the earth and the moon. Consequently, it is impossible to exert time-sensitive control based on collecting global information about the network. If we allow to spend sufficient time to collect global information, the information so gathered is too old to apply to time-sensitive control. So, in a high-speed network, the frameworks of time-sensitive control are inevitably autonomous decentralized systems [1, 2, 3].

This paper focuses on back-pressure type flow control in networks, in which nodes handle their local traffic flow themselves based only on the information they are aware of. It is, of course, preferable that the decision making of each node leads to high performance for the whole network. In flow control, we use the total throughput of a network as a global performance measure [1]. We investigate the behavior of local packet flow and the global performance measure when a node is congested, and discuss an appropriate flow control model on the basis of simulation results. In addition, we investigate the stability and adaptability of the network

Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)
0-7695-1876-1/03 $17.00 © 2003 IEEE

Table 1. Classification of flow control mechanisms with respect to collecting information.

	low-speed NW	high-speed NW
centralized control	1-A	1-B
decentralized control	1-C	1-D

Table 2. Classification of decentralized flow control mechanisms with respect to control delay requirement.

	decision-making	
	long time-scale	short time-scale
controlled by end hosts (end-to-end, end-to-node)	2-A	2-B
controlled by nodes (node-by-node)	2-C	2-D

performance when the capacity of a link is changed.

This paper is organized as follows. In Section 2, we discuss related works, comparing them with our work by categorizing flow control mechanisms. In Section 3, we describe a performance measure for the whole network and the principle of our flow control model. In Section 4, we show two types of autonomous decentralized flow control. In Section 5, we describe two simulation models, and the corresponding results is shown in Section 6. Finally, we conclude this paper in Section 7.

2 Related Works

In general, the technique used for flow control for high-speed networks should satisfy the following requirements:

1. With regard to the collection of information: it must be possible to collect the information used in the control.

2. With regard to the delay in applying control: the control should take effect immediately.

In high-speed networks, we cannot collect global information about the network. So, let us classify the flow control mechanisms with respect to collecting information as shown in Table 1.

Centralized control requires the collection of global information about the network, but this is impossible in high-speed networks. Therefore, class 1-B control mechanism cannot be realized. In low speed networks, both classes 1-A and 1-C are possible. There are many papers which consider these classes. They mainly study the optimization of flow control problems in a framework of solving linear programs. Techniques for addressing rate control, bandwidth assignment, deadlock resolution, resource allocation or flow fairness problems for each source by optimizing some end-to-end utility function have been studied in [4, 5, 6, 7, 8]. [9] studied the issue of to reducing the convergence time in solving optimization problems.

Our target is flow control in a high-speed network, for which the framework is inevitably autonomous decentralized control. So, we focus on class 1-D.

We can again classify the decentralized flow control mechanisms, from the point-of-view of control delay requirement as shown in Table 2.

Flow control by end hosts including TCP is widely used in the current networks, and there is much research in this area.

Based on the optimization problem, [10] introduces a decentralized marking mechanism for early notification of congestion. For window-based flow control, the optimization problems of some aggregated utility functions have been studied in [11]. Solving these optimization problems, however, requires enough time to be available for calculation, and so it is difficult to apply them to decision-making in a very short time-scale.

Johari and Tan [12] studied stable end-to-end congestion control when the propagation delay is large relative to the queueing delay. Each end system requires knowledge only of its own round-trip delay. [13] studied rate control and window control with feedback from nodes to end hosts. Each feedback indicates the state of the buffer at the node — whether it is above or below a threshold. Since the control by end hosts cannot be applied to decision-making in a time-scale shorter than the round-trip delay, it is inadequate for application to decision-making in very short time-scale. Therefore, these methods are categorized as class 2-A. In high-speed networks, due to the fact that many packets are influenced by control delay, very short control delay is required. So class 2-B cannot be realized. Our target is node-by-node control and is categorized as class 2-D.

Bartal *et al.* [4] studied global optimization of flow control using local information. The motivation of their work was to enable the distributed routers in high-speed networks to make decisions on flow control as quickly as possible, and they studied the problem in a framework of solving linear programs by distributed agents. Though this motivation is similar to that of our work, their study assumes the distributed agents can obtain detailed information about networks if we allow them to spend sufficient time. As stated above, our standpoint is based on the fact that we cannot obtain detailed, useful and up-to-date information about the whole network in a high-speed network environment, even if we do not limit the time that we can take.

In our previous works, we have investigated the characteristics of autonomous decentralized flow control in a high-speed network [1, 2, 3]. We proposed a simple and effective method of flow control in [2]. Since the proposed control uses less information than the control methods described in [1, 3], the proposed control is relatively simple. In this

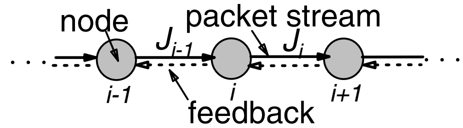

Figure 3. Network model

paper, by changing the capacity of a link, we compare autonomous decentralized flow control mechanisms with respect to the stability and adaptability of the network performance.

3 Models

3.1 Performance Measure

Each packet in a network is either in a node or on a link. Since the packets currently stored in nodes are not being transmitted over the network, it is natural to define the total throughput of the network as a global performance measure as follows. We define the total throughput of a network at time t as the amount of data being propagated on the network [1, 2, 3, 14]. In other words, it is the number of packets being propagated on all links in the network at time t.

On the other hand, the only packets we can control are those stored in nodes, and not those being propagated. Thus, higher performance of the whole network involves many uncontrollable packets being propagated on links. Therefore, inappropriate flow control cannot produce a state that has high performance and stability.

3.2 Network and Flow Control Models

Our network model has a simple 1-dimensional configuration (Fig. 3). All nodes have two incoming links and two outgoing ones for a one-way packet stream and feedback information, that is, node i ($i = 0, 1, 2, \ldots$) transfers packets to node $i+1$ and node $i+1$ sends feedback information (node information) to node i. For simplicity, we assume that packets have a fixed length in bits.

All nodes are capable of receiving and sending node information from/to adjacent downstream and upstream nodes, respectively. Each node i can receive node information sent from the downstream node $i+1$, and can send its node information to the upstream node $i-1$. When node i receives node information from downstream node $i+1$, it determines the transmission rate for packets to the downstream node $i+1$ using the received node information and adjusts its transmission rate towards the downstream node $i+1$. The framework of node behavior and flow control is summarized as follows:

- Each node i autonomously determines the transmission rate J_i based only on information it is aware of, *i.e.*, the node information obtained from the downstream node $i+1$ and its own node information.
- The rule for determining the transmission rate is the same for all nodes.
- Each node i adjusts its transmission rate towards the downstream node $i+1$ to J_i .
(If there are no packets in node i, the packet transmission rate is 0.)
- Each node i autonomously creates node information according to a predefined rule and sends it to the upstream node $i-1$.
- The rule for creating the node information is the same for all nodes.
- Packets and node information both experience the same propagation delay.

As mentioned above, the framework of our flow control model involves both autonomous decision-making by each node and interaction between adjacent nodes. There is no centralized control mechanism in the network. More precisely, it is impossible to achieve centralized control in a high-speed network environment. Hereafter, we investigate the behavior of the total network performance driven by two different flow control mechanisms, applied for different processes used to determine the transmission rate.

4 Preliminary Description of Flow Control

4.1 Packet Flow

In this paper, we focus on the stability and adaptability of flow control in the congested state, and we consider packet flow in a heavy-traffic environment. In this situation, we let the packet flow be J_i if the transmission rate specified by node i is J_i. This is because node i has sufficient packets to transfer. Hereafter, we identify the packet flow with the transmission rate specified by the node.

We define the packet flow as

$$J_i(t) := \alpha\, r_i(t-d_i) - D\,(n_{i+1}(t-d_i) - n_i(t)), \quad (1)$$

where $n_i(t)$ denotes the number of packets in node i at time t, r_i is the rate sent by the downstream node $i+1$ as node information, α (>0) and D (>0) are constants (we call α and D the drift and diffusion coefficients, respectively), and d_i denotes the propagation delay between node i and node $i+1$. In addition, $(r_i(t-d_i), n_{i+1}(t-d_i))$ is notified from the downstream node $i+1$ with the propagation delay d_i. We call the first and second terms on the right hand side of Eq. (1) the drift and diffusion terms, respectively.

If there is no packet loss in the network, the temporal variation of $n_i(t)$ is expressed as

$$n_i(t+\epsilon) - n_i(t) = \epsilon\,[J_{i-1}(t-d_{i-1}) - J_i(t)]\,, \quad (2)$$

where $\epsilon > 0$ is a small number.

To estimate the temporal variation roughly, we replace i with x and apply continuous approximation. Then the propagation delay becomes $d_i \to 0$ for all i and the packet flow is expressed as

$$J(x,t) = \alpha\, r(x,t) - D\,\frac{\partial n(x,t)}{\partial x}, \qquad (3)$$

and the temporal variation of the number of packets at x is expressed as a diffusion type equation,

$$\frac{\partial n(x,t)}{\partial t} = -\alpha\,\frac{\partial r(x,t)}{\partial x} + D\,\frac{\partial^2 n(x,t)}{\partial x^2}, \qquad (4)$$

by using the continuous equation

$$\frac{\partial n(x,t)}{\partial t} + \frac{\partial J(x,t)}{\partial x} = 0. \qquad (5)$$

That is, our method aims to perform flow control using the analogy of a diffusion phenomenon.

Hereafter, we consider two types of flow control and compare them. One type handles the drift term and the other controls both the drift and diffusion terms.

4.2 Drift-Type Flow Control and Stability

In this subsection, we set $D = 0$ in Eqs. (1) and (3), and investigate the characteristics of a flow control mechanism whose packet flow is determined only by the drift term.

Let the number of packets in the network be N. To obtain higher network performance, flow control should enable a state in which many packets are being propagated on links. This state corresponds to a state in which there are fewer packets in nodes.

The simplest strategy for achieving this state is for each node to attempt to decrease the number of packets in it. Therefore, the temporal variation of $n_i(t)$ should be

$$n_i(t+\epsilon) - n_i(t) < 0. \qquad (6)$$

From Eq. (2) and $D = 0$, this strategy means that node i notifies a smaller rate to the upstream node $i-1$ than the rate notified by the downstream node,

$$r_{i-1}(t) < r_i(t-d_i). \qquad (7)$$

However, if all nodes use this strategy, then the total throughput decreases with time as a result. Therefore, the strategy described by Eq. (7) cannot be used continuously.

Conversely, if we use the rate specified to the upstream node as

$$r_{i-1}(t) > r_i(t-d_i), \qquad (8)$$

then $n_i(t)$ increases with respect to time (when there are many packets in the upstream node). But the buffer in each node has a finite capacity, so this strategy described by Eq. (8) cannot be used continuously either.

If we set the rate specified to the upstream node as

$$r_{i-1}(t) = r_i(t-d_i), \qquad (9)$$

then $n_i(t)$ does not change with respect to time under a heavy traffic condition. This means that the strategy described by Eq. (9) does not diminish the total performance of the network. However, when some node is congested, its restoration requires a long time. Thus, the strategy described by Eq. (9) can also not be used continuously.

From the above considerations, we choose the following strategy. The rate specified from node i to the upstream node $i-1$ is determined according to the state of node i. Let the objective of n_i be n_s. If $n_i(t) > n_s$, then r_{i-1} is specified by using Eq. (7); if $n_i(t) < n_s$, then r_{i-1} is specified by using Eq. (8); and if $n_i(t) = n_s$, then r_{i-1} is specified by using Eq. (9).

Since the above flow control does not use the diffusion term, we call it drift-type flow control in this paper.

4.3 Diffusion-Type Flow Control and Stability

In this subsection, we set $D > 0$ in Eqs. (1) and (3), and investigate the characteristics of the flow control mechanism whose packet flow is determined by both drift and diffusion terms.

In this control mechanism, node i's packet transmission rate to the downstream node $i+1$ is determined as

$$J_i(t) = \alpha\, r_i(t-d_i) - D\,(n_{i+1}(t-d_i) - n_i(t)), \qquad (10)$$

and the node information of node i sent to the upstream node $i-1$ is determined as

$$r_{i-1}(t) = J_i(t). \qquad (11)$$

In the case where $D = 0$ and $\alpha = 1$, Eqs. (10) and (11) reduce to a drift-type flow control specified by Eq. (9). In the framework of Eqs. (10) and (11), the node information of i specified to the upstream node $i-1$ is a pair of values $(r_{i-1}(t),\, n_i(t))$.

Since the above flow control uses the diffusion term, we call it the diffusion-type flow control in this paper.

5 Simulation Model

In this section, we consider a simple network model with a bottleneck link having a narrow bandwidth and compare the performance of the two different flow control principles described in the previous sections.

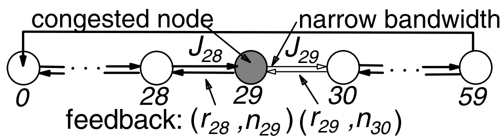

Figure 4. Network model with a bottleneck link

5.1 Network Model

Figure 4 shows our network model, which is a closed network with a 1-dimensional configuration and toroidal boundary. The network has a congested node and a bottleneck link. All the other nodes and links are in the same condition. This model simulates the situation when congestion occurs at a certain node. We are interested in the behavior of the local congestion, that is, whether:

- it causes deterioration of the total network performance through interaction among nodes, or
- it diminishes with time.

Detailed conditions of our network model are listed below.

- Number of nodes: $m = 60$. Each node specified by i $(\mathrm{mod}\ 60)$.
- Propagation delay between adjacent nodes: 1 (unit time)
- Index of the congested node: $i = 29$
- Total number of packets in the network: $N = 6000$
- Maximum number of packets on a link (except the bottleneck link): $L_c = 100$
- Maximum number of packets on the bottleneck link (between nodes $i = 29$ and 30): $L_b = 25$, 50, or 75 (that is, 1/4, 1/2, or 3/4 of the bandwidth of other links, and the same length)

To investigate the stability under congestion, in addition to the above conditions, we set the initial condition for congested node $i = 29$ as follows.

- Number of packets in node $i = 29$ at time $t = 0$: 400
- The other 5600 packets are randomly configured in other nodes and on other links.

5.2 Drift- and Diffusion-Type Flow Control Mechanisms

As a model for the drift-type flow control, we set an objective for the number of packets in a node to be $n_s = 60$, and set the following transmission rate and node information.

$$J_i = \min(r_i, L_i), \tag{12}$$

$$r_{i-1} = \begin{cases} J_i \times 0.9 & (n_i > n_s), \\ J_i \times 1.0 & (n_i = n_s), \\ J_i \times 1.1 \cdots & (n_i < n_s), \end{cases} \tag{13}$$

where L_i denotes the link capacity between nodes i and $i+1$.

As a model for the diffusion-type flow control, we set $D = 0.1$ in Eqs. (10) and (11), and use the following flow control model.

$$J_i = \begin{cases} L_i, & (\tilde{J}_i > L_i), \\ 0, & (\tilde{J}_i < 0), \\ \tilde{J}_i, & (\text{otherwise}), \end{cases} \tag{14}$$

$$r_{i-1} = J_i, \tag{15}$$

where $\tilde{J}_i = \alpha\, r_i - D\,(n_{i+1} - n_i)$ and α is a constant.

6 Simulation Results: Stability and Adaptability

From the simulation results for the drift- and diffusion-type control models, we compare the total throughput of the network. In addition, we discuss the stability and adaptability of the both types of flow control model through the observation of the total throughput.

6.1 Stability and Adaptability in the Case of the Appearance of a Bottleneck Link

This subsection considers the case where the capacity of a link in the network is suddenly reduced to a narrow bandwidth. No node is aware of the change of the link state and new capacity of the link. We investigate the stability and adaptability of the drift- and diffusion-type control models through the observation of the total throughput of the network.

Figure 5 shows the simulation result for the drift-type flow control model when $L_b = 50$, *i.e.*, a half of the other link capacities. The horizontal axis of each graph denotes node ID and the vertical axis denotes the number of packets stored in the node. In addition, t denotes the simulation time and initially $t = 0$.

The number of packets in the congested node $i = 29$ decreases with time, and nodes that store packets have around 100 packets at $t = 100$ and 200. Each node has the objective $n_s = 60$, so the strategy of each node is a failure as a result.

Similarly, Fig. 6 shows the simulation result for the diffusion-type flow control model under the same conditions as described for Fig. 5. We chose parameters as

Proceedings of the Sixth International Symposium on Autonomous Decentralized Systems (ISADS'03)
0-7695-1876-1/03 $17.00 © 2003 IEEE

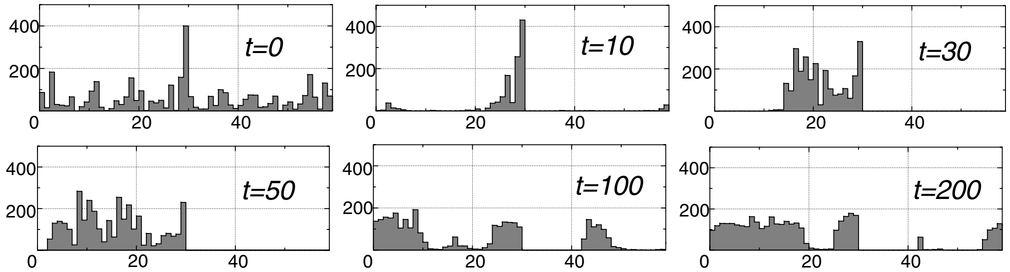

Figure 5. Temporal variation of the number of packets in each node for the drift-type flow control mechanism ($n_s = 60$)

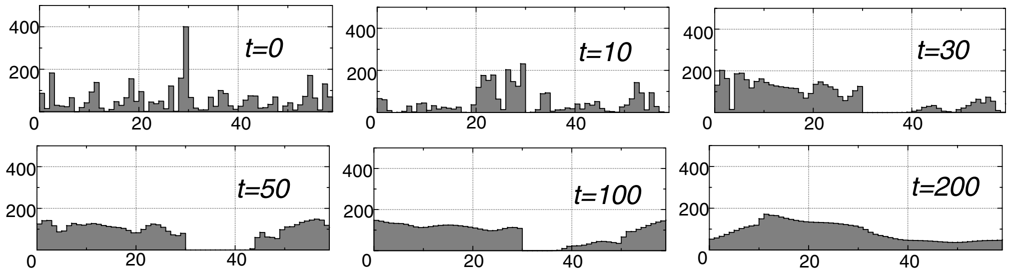

Figure 6. Temporal variation of the number of packets in each node for the diffusion-type flow control mechanism ($\alpha = 1.0$, $D = 0.1$)

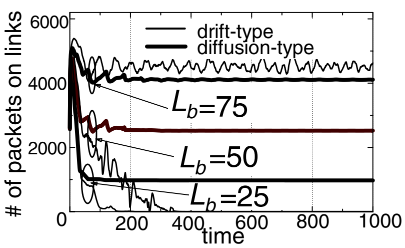

Figure 7. Temporal behavior and stability of the total throughput of the network for two different flow control mechanisms

$\alpha = 1.0$ and $D = 0.1$. The number of packets in congested node $i = 29$ decreases with time, and the distribution of the number of the packets stored in nodes is smoothly distributed over the network at $t = 200$.

Figure 7 shows the total throughput for both types of flow control model. The horizontal axis denotes the simulation time and the vertical axis denotes the total throughput (*i.e.*, the total number of packet being propagated on links). The simulation conditions are the same as for the cases illustrated in Figs. 5 and 6 and we show the cases of three different capacities of the bottleneck link, $L_b = 25, 50,$ and 75. In the case where $L_b = 75$, the drift-type flow control achieves higher total throughput than that obtained from the diffusion-type flow control. However, in the other cases, the drift-type fails to control the total throughput, which falls to zero with time. On the other hand, diffusion-type flow control achieves stable total throughput of the network. It is remarkable that stability is achieved irrespective of L_b.

Next, we discuss the results from a quantitative point of view. Let us compare both types of control model in the case where $L_b = 50$. This is the same condition as the result shown in Figs. 5 and 6.

For the drift-type control models, the total throughput decreases with time. This means that the flow control model inappropriately influences the global performance of the network. If all nodes achieve their objective of $n_s = 60$, the total throughput should be 2400 (total of 6000 packets; and (60 packets/node $\times$ 60 nodes) packets stored in the nodes). On the other hand, for the diffusion-type control model, the total throughput decreases with time but becomes stable around 2400. From the link capacity of the bottleneck link $L_b = 50$, the maximum value of the sustainable total throughput (the maximum number of packets being propagated stably on links) is 3000, *i.e.*, 50 packets/link $\times$ 60 links. Thus, the diffusion-type flow control achieves 80% of the maximum value of the total throughput and its value is stable.

If we can choose an appropriate value of the objective n_s for the drift-type flow control, the total throughput may be stable and adaptive. The case where $L_b = 75$ implies that this is feasible. However, the value of n_s should depend on the bandwidth of the bottleneck link. Since nodes cannot be aware of information about the bandwidth in a high-speed network environment, the drift-type control cannot achieve high performance. In the diffusion-type control model, although no node is aware of the bandwidth of the bottleneck link, stable and high performance is achieved.

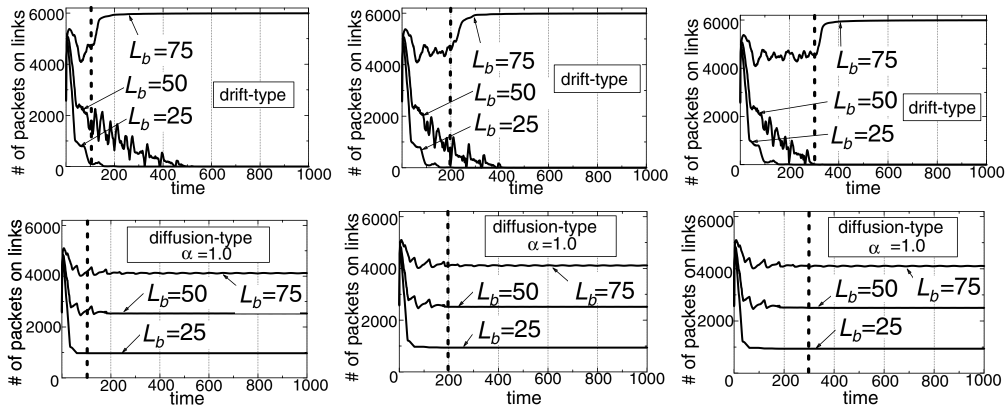

Figure 8. Temporal behavior and stability of the total throughput of the network for two different flow control mechanisms and three different times ($t = 100$, 200, and 300) at which the bottleneck is restored

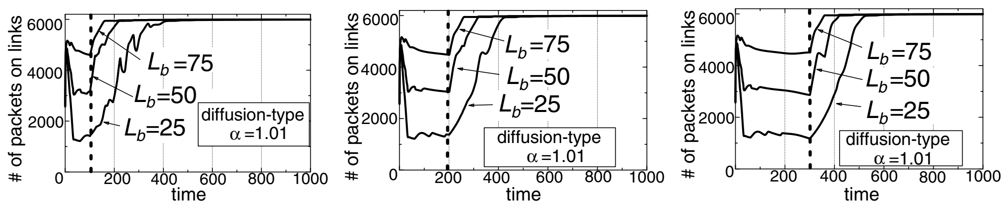

Figure 9. Temporal behavior and stability of the total throughput for the diffusion-type flow control mechanism with $\alpha = 1.01$ when the bottleneck is restored ($t = 100$, 200, and 300)

6.2 Stability and Adaptability in the Case of Restoration of a Bottleneck Link

This subsection considers the situation where the capacity of the bottleneck link is suddenly restored. No node is aware of the change of the link state and the restored capacity of the link. We investigate the stability and adaptability of both types of control through observation of the total throughput of the network.

Figure 8 shows the total throughput in the case where the capacity of the bottleneck link L_b is restored to 100 at time $t = 100$, 200, and 300 for both models. The horizontal axis denotes the simulation time and the vertical axis denotes the total throughput. The broken lines in these figures indicate the time when L_b is restored. The top three figures show the results for drift-type control and the bottom three those for diffusion-type control. The simulation conditions are the same as the case for the previous subsection except for the restoration of the bottleneck link. We show the cases of three different capacities of the bottleneck link, $L_b = 25$, 50, and 75.

In the case where $L_b = 75$, the drift-type flow control restores high total throughput of around 6000. It is independent of the time when the restoration occurs. However, for other initial capacities, the drift-type fails to control the total throughput, which falls to zero.

On the other hand, the diffusion-type flow control achieves stable total throughput of the network. However, although the capacity of the bottleneck link is restored, the total throughput is not restored in any of the cases. This is because we set the drift coefficient of $\alpha = 1.0$ for the diffusion-type flow control. This setting of the drift coefficient is derived from Eq. (9) for balancing input and output flows.

In order to realize flow control which has both stability and adaptability, we choose the diffusion-type control with $\alpha > 1.0$. Figure 9 shows the total throughput in the case where the capacity of the bottleneck link is restored to 100 at time $t = 100$, 200, and 300 for the diffusion model with $\alpha = 1.01$. From these figures, the diffusion-type control with $\alpha > 1.0$ may be seen to have adaptability for the restoration of the link capacity.

6.3 Stability and Adaptability in the Case where a Bottleneck Appears and Recovers Repeatedly

This subsection considers the situation where the capacity of the bottleneck link alternates between normal and restricted values. No node is aware of the changes in the link state. We investigate the stability and adaptability of the diffusion-type control with the drift coefficient of $\alpha > 1$ through observation of the temporal behavior of the total throughput of the network.

Figure 10 shows the total throughput in case where the bottleneck appears and recovers repeatedly for the diffusion-type control with $\alpha = 1.01$. The capacities of the

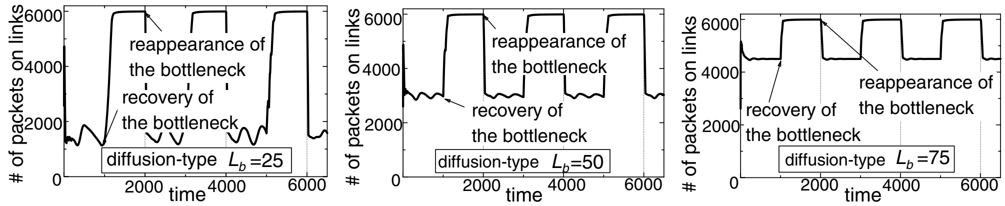

Figure 10. Temporal behavior and stability of the total throughput for the diffusion-type flow control mechanism with $\alpha = 1.01$ when the capacity of the bottleneck is alternately restricted and restored

bottleneck links are $L_b = 25, 50$, and 75, respectively. The horizontal axis denotes the simulation time and the vertical axis denotes the total throughput.

The period of the bottleneck link being alternately changed between appearance and restoration is 1000. Simulation conditions are the same as the case for the previous subsection except for the state of the bottleneck link. In all cases, the diffusion-type control with $\alpha = 1.01$ achieves the adaptable global performance of the network.

7 Conclusions

This paper has presented a framework for flow control in high-speed networks as an autonomous decentralized system. We have showed two typical back-pressure type flow control models based on the framework. The drift-type flow control handles the drift term of the packet flow and the diffusion-type flow control handles both the drift and diffusion terms of the packet flow. For both types of control, nodes handle their local traffic flow themselves based only on the information they are aware of.

To investigate the behavior of local packet flow and the global performance measure when a node is congested, we compared two models through simulations. For comparison, we used the total throughput as the flow control performance measure.

Although the drift-type control cannot achieve high performance adaptively, the diffusion-type does achieve stable performance in congested situations. In particular, the diffusion-type control with a drift coefficient of $\alpha > 1$ achieves high performance adaptively, even in a situation in which the congested state changes dynamically.

We are interested in the appropriate values of the drift and diffusion coefficients, α and D. These issues will be the subject of further study.

References

[1] M. Aida and K. Horikawa. Stability analysis for global performance of flow control in high-speed networks based on statistical physics. *IEICE Transactions on Communications*, E82-B(12):2095–2106, December 1999.

[2] M. Aida and C. Takano. Stability of autonomous decentralized flow control schemes in high-speed networks. In *Proc. of IEEE ICDCS 2002 Workshop (ADSN 2002)*, pages 63–68, 2002.

[3] K. Horikawa, M. Aida, and T. Sugawara. Traffic control scheme under the communication delay of high-speed networks. In *Proc. of Int. Conf. on Multi-Agent Systems (ICMAS) '96*, pages 111–117, 1996.

[4] Y. Bartal, J. Byers, and D. Raz. Global optimization using local information with applications to flow control. In *Proc. of the 38th Ann. IEEE Symp. on Foundations of Computer Science*, October 1997.

[5] B. Awerbuch and Y. Azar. Local optimization of global objectives: competitive distributed deadlock resolution and resource allocation. In *Proc. of the 35th Ann. IEEE Symp. on Foundations of Computer Science*, pages 240–249, 1994.

[6] D. Lapsley and S. Low. An IP implementation of optimization flow control. In *Proc. of IEEE GLOBECOM '98*, November 1998.

[7] S. H. Low and D. E. Lapsley. Optimization flow control-I: basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[8] K. Kar, S. Sarkar, and L. Tassiulas. A simple rate control algorithm for maximizing total user utility. In *Proc. of IEEE INFOCOM 2001*, pages 133–141, 2001.

[9] B. Awerbuch and Y. Shavitt. Converging to approximated max-min flow fairness in Logarithmic Time. In *Proc. of IEEE INFOCOM '98*, pages 1350–1377, March 1990.

[10] S. K. R. Srikant. A decentralized adaptive ECN marking algorithm. `citeseer.nj.nec.com/487389.html`

[11] J. Mo and J. Walrand. Fair end-to-end window based congestion control. *IEEE/ACM Trans. Networking*, 8(5):556–567, October 1999.

[12] R. Johari and D. Tan. End-to-end congestion control for the Internet: Delays and stability. 2001. `citeseer.nj.nec.com/johari01endtoend.html`

[13] F. Bonomi, D. Mitra, and J. B. Seery. Adaptive algorithms for feedback-based flow control in high-speed, wide-area ATM networks. In *IEEE Journal on Selected Areas in Communications*, 13:1267–1283.

[14] D. Mitra. Optimal design of windows for high speed data networks. In *Proc. of IEEE INFOCOM '90*, pages 1156–1163, 1990.